


MS150750.01/MSFTP159US

AF IPW
2124CERTIFICATE OF MAILING

I hereby certify that this correspondence (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: **Mail Stop Appeal Briefs – Patents**, Assistant Commissioner for Patents, U.S. Patent and Trademark Office, Washington, D.C. 20231.

Date:

10-25-04


Himanshu S. Amin**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re patent application of:

Applicant(s): Joseph A. Porkka

Serial No: 09/729,448

Filing Date: December 4, 2000

Examiner: Trenton J. Roche

Art Unit: 2124

Title: SYSTEM AND METHOD TO COMMUNICATE, COLLECT AND
DISTRIBUTE GENERATED SHARED FILES

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

10/28/2004 HALI11 00000026 09729448

APPEAL BRIEF

01 FC:1402

340.00 OP

Dear Sir:

Applicant submits this brief in connection with an appeal of the above-identified patent application. A credit card payment form is filed concurrently herewith in connection with all fees due regarding this appeal brief. In the event any additional fees may be due and/or are not covered by the credit card, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1063 [MSFTP159US].

I. Real Party in Interest (37 C.F.R. §41.37(c)(1)(i))

The real party in interest in the present appeal is Microsoft Corporation, the assignee of the present application.

II. Related Appeals and Interferences (37 C.F.R. §41.37(c)(1)(ii))

Appellant, appellant's legal representative, and/or the assignee of the present application are not aware of any appeals or interferences which may be related to, will directly affect, or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims (37 C.F.R. §41.37(c)(1)(iii))

Claims 5, 6, and 26-35 have been withdrawn. Claims 1-4 and 7-25 stand rejected by the Examiner. The rejection of claims 1-4 and 7-25 is being appealed.

IV. Status of Amendments (37 C.F.R. §41.37(c)(1)(iv))

No claim amendments have been entered after the Final Office Action.

V. Summary of Claimed Subject Matter (37 C.F.R. §41.37(c)(1)(v))**A. Independent Claim 1**

Independent claim 1 relates to a system for building a software system. The building system includes a first component for building a list of file names of published files to be shared by a plurality of build machines employed in building the software system. (*See e.g.*, Application at p. 3, lines 6-9). The system also includes a second component for distributing to one or more of the build machines one or more published files, identified in the list of file names, that are to be stored persistently by the one or more build machines. (*See e.g.*, Application at p. 3, lines 10-12).

B. Independent Claim 15

Independent claim 15 relates to a system for building a software system employing a plurality of build machines. The building system includes a component for

broadcasting to one or more of the build machines. (*See e.g.*, Application at p. 3, lines 14-16). The component broadcasts one or more published build files, which are stored persistently by the set of build machines. (*See e.g.*, Application at p. 3, lines 14-16). The system also includes a component for selectively receiving and storing persistently one or more of the broadcast published build files. (*See e.g.*, Application at p. 3, lines 16-18).

C. Independent Claim 19

Independent claim 19 relates to a method for building a software system. The method includes collecting one or more file names from one or more build machines. (*See e.g.*, Application at p. 3, lines 20-21). The file name(s) correspond to one or more build files. (*See e.g.*, Application at p. 3, line 21). The method also includes determining which of the build file(s) the build machine(s) will send to a post build machine. (*See e.g.*, Application at p. 3, lines 21-22). The method provides for persistently storing the build file(s) on the build machine(s). (*See e.g.*, Application at p. 3, line 23). The method further includes sending to the build machine(s) a list of file names of build files to be sent to the post build machine and sending the build file(s) to the post build machine. (*See e.g.*, Application at p. 3, lines 24-25). The method also provides for determining which of the build file(s) the build machine(s) will receive from the post build machine, and sending a list of file names of build files to be received from the post build machine to the build machine(s). (*See e.g.*, Application at p. 3, lines 26-28). The method further includes distributing the build file(s) to the build machine(s) and persistently storing the build file(s) distributed to the build machine(s). (*See e.g.*, Application at p. 3, lines 28-30).

D. Independent Claim 20

Independent claim 20 relates to a computer-readable medium having computer-executable instructions for building a software system. (*See e.g.*, Application at p. 3, lines 19-30). The computer-executable instructions include collecting one or more file names from one or more build machines. The file name(s) correspond to one or more build files. The computer-executable instructions also include determining which of the build file(s) the build machine(s) will send to a post build machine. The computer-

executable instructions also include persistently storing the build file(s) on the build machine(s). The computer-executable instructions also include sending to the build machine(s) a list of file names of build files to be sent to the post build machine and sending the build file(s) to the post build machine. The computer-executable instructions provide for determining which of the build file(s) the build machine(s) will receive from the post build machine, and sending a list of file names of build files to be received from the post build machine to the build machine(s). The computer-executable instructions also provide for distributing the build file(s) to the build machine(s) and persistently storing the build file(s) distributed to the build machine(s).

E. Independent Claim 21

Independent claim 21 relates to a method for building a software system. (*See e.g.*, Application at p. 10, line 22 through p. 12, line 11). The method includes collecting one or more build files from one or more build machines. The method further includes distributing the build file(s) to the build machine(s).

F. Independent Claim 24

Independent claim 24 relates to a computer-readable medium having computer-executable instructions. (*See e.g.*, Application at p. 10, line 22 through p. 12, line 11). The computer-executable instructions include instructions for collecting one or more build files from one or more build machines and distributing the build file(s) to the build machine(s). The computer-executable instructions also include determining which of the transmitted build files to store persistently, and persistently storing one or more of the transmitted files.

G. Independent Claim 25

Independent claim 25 relates to a system for building a software system. The system includes means for collecting from one or more build machines, one or more file names corresponding to the one or more build files. (*See e.g.*, Application at p. 7, lines 12-19; p. 8, lines 23-29; p. 9, lines 20-31; p. 10, lines 23-31; p. 11, lines 1-7 and 21-31). The system also includes means for determining which of the one or more build files the

one or more build machines are to send to a post build machine. (*See e.g.*, Application at p. 7, lines 17-19; p. 8, lines 23-31; p. 9, lines 20-31; p. 10, lines 29-31; p. 11, lines 1-7 and 21-31; p. 12, lines 20-31; p. 13, lines 1-11). The system also includes means for sending the one or more build files to the post build machine. (*See e.g.*, Application at p. 7, lines 26-31; p. 8, lines 23-31; p. 9, lines 20-25; p. 11, lines 8-14 and 26-27; p. 12, lines 2-5). The system also includes means for determining which of the one or more build files the one or more build machines are to receive from the post build machine. (*See e.g.*, Application at p. 7, lines 23-26; p. 8, lines 23-31; p. 9, lines 20-25; p. 11, lines 8-14 and 26-27; p. 12, lines 4-5 and lines 20-31; p. 13, lines 1-11). The system further includes means for sending to the one or more build machines a list of file names of build files to be sent to the one or more build machines and the build files to be received from the one or more build machines. (*See e.g.*, Application at p. 7, lines 26-31; p. 8, lines 23-30; p. 9, lines 20-31; p. 10, lines 29-31; p. 11, lines 1-7 and 25-31; p. 12, lines 1-8). The system also provides for means for distributing one or more build files to the one or more build machines. (*See e.g.*, Application at p. 7, lines 26-31; p. 8, lines 23-31; p. 9, lines 20-31; p. 10, lines 23-31; p. 11, lines 1-7 and 25-31; p. 12, lines 1-3). The system also includes means for persistently storing one or more build files distributed to the one or more build machines. (*See e.g.*, Application at p. 8, lines 14-15 and 30-31; p. 10, lines 4-7; p. 11, lines 17-18; p. 12, lines 8-9).

The “means for” limitations described above are identified as limitations subject to the provisions of 35 U.S.C. §112 ¶6. The corresponding structures are identified with reference to the specification and drawings in the parentheticals above corresponding to those claim limitations.

VI. Grounds of Rejection to be Reviewed (37 C.F.R. §41.37(c)(1)(vi))

A. Whether claims 1-4, 7-13 and 15-25 are unpatentable under 35 U.S.C. §102(e) over Boehm *et al.* (U.S. 6,457,170) ; and

B. Whether claim 14 is unpatentable under 35 U.S.C. §103(a) over Boehm *et al.* in view of Lubkin *et al.* (U.S. 5,339,435).

VII. Argument (37 C.F.R. §41.37(c)(1)(vii))

The subject claims relate to systems, methods, and computer-readable media for managing the storage, collection, and distribution of files that are used to build a software system. (See *e.g.*, Application at p. 25, lines 2-10). In accordance with applicant's claimed invention, build files are persistently stored on the build machine and distributed (or broadcast) as necessary to build a software system. For example, the files can be distributed (or broadcast) to a post build machine or between build machines. (See *e.g.*, Application at p. 1, lines 26-31; p. 8, lines 26-31; p. 9, lines 23-31; p. 10, lines 3-6 and lines 14-17; p. 11, lines 13-17 and 26-28; p. 12, lines 4-7). Because the build files are persistently stored on the build machines, the repetitive downloads that occur in systems in which the build files are stored on a central server are eliminated. (See *e.g.*, p. 8, lines 14-17; p. 10, lines 4-7 and 14-17; p. 11, lines 17-20; p. 12, lines 8-11). Thus, the invention as recited in the subject claims allow for a more efficient and faster software build than could be accomplished if the build files were not persistently stored on the build machines.

A. Rejection of Claims 1-4, 7-13 and 15-25 Under 35 U.S.C. §102(e)

i. Claims 1-4 and 7-13

Claims 1-4 and 7-13 are argued separately from the other claims. Claims 1-4 and 7-13 stand rejected under 35 U.S.C. §102(e) as being anticipated by Boehm *et al.* Reversal of this rejection is respectfully requested for at least the following reasons. Neither the preferred (multi-workstation) embodiment nor the alternative (single workstation) embodiment taught by Boehm *et al.* discloses each and every limitation of the subject claims. In particular, the multi-workstation embodiment taught by Boehm *et al.* does not disclose a system having a second component for distributing to one or more build machines one or more published files, identified in the list of file names, that are to be *stored persistently by the one or more build machines* as in applicants' claimed invention. Moreover, the single workstation embodiment taught by Boehm *et al.* does

not have a first component for building a list of file names of published files to be *shared by a plurality of build machines employed in building the software system*. Boehm *et al.* does not teach or suggest such claimed features of applicant's invention.

A single prior art reference anticipates a patent claim if “*each* and *every* limitation set forth in the patent claim” is disclosed either expressly or inherently. (*Trintec Industries, Inc. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 1295, 63 U.S.P.Q.2d 1597, 1599 (Fed. Cir. 2002) (citing to *Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1052-53 (Fed. Cir. 1987))) (emphasis added). Moreover, “[t]he *identical* invention must be shown in as *complete* detail as is contained in the patent claim.” (*Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989) (citing *Jamesbury Corp. v. Litton Industrial Products, Inc.*, 756 F.2d 1556, 1560, 225 U.S.P.Q. 253, 257 (Fed. Cir. 1985); and *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 1548, 220 U.S.P.Q. 193, 198 (Fed. Cir. 1983))) (emphasis added).

Boehm *et al.* discloses a system for building software in a networked software development environment. In the preferred (multi-workstation) embodiment of Boehm *et al.*, the source and object files are stored in a network cache memory that is not resident on the user's local workstation. (See *e.g.*, col. 6, lines 2-14). As defined in Boehm *et al.*, a network cache memory is an “electronic memory 32 *located on the network* that is quickly accessible by each local workstation.” (See col. 5, lines 39-41; *see also* Figure 1) (emphasis added). To build a software program, a user must first create a build list by entering the name of each source and object file constituent of the software to be built. (See col. 2, lines 43-47; col. 4, lines 6-10; and col. 18, line 36). From this build list, the system creates a cache link structure (*i.e.*, a directory tree of source and object file links). The system provides the cache link structure to a software building program, such as MAKE, running on the user's local workstation. The software building program accesses the files stored in the network cache memory by using the links to the cached copies of the files provided in the cache link structure. (See *e.g.*, col. 2, lines 46-64 and col. 14, lines 51-55). Boehm *et al.* also discloses an alternative (single workstation) embodiment

that runs on a standalone workstation and that utilizes a local cache to store files. (See e.g., col. 9, lines 12-14).

None of the embodiments of Boehm *et al.* disclose each and every limitation recited in independent claim 1. Independent claim 1 recites a system having a second component for distributing to one or more build machines one or more published files, identified in the list of file names, that are to be ***stored persistently by the one or more build machines***. As discussed *supra*, in the multi-workstation embodiment of Boehm *et al.*, the software building program is resident on the user's local workstation, and builds the software from source and object files that are stored *in a network cache memory* that is not resident on the user's local workstation. (See e.g., Boehm *et al.* at Fig. 1 and col. 5, lines 39-41). Thus, in the multi-workstation embodiment disclosed in Boehm *et al.*, the build files are not stored on the build machines.

Similarly, the alternative (single workstation) embodiment disclosed by Boehm *et al.* does not disclose each and every limitation of claim 1. Independent claim 1 recites ***a system having a first component for building a list of file names of published files to be shared by a plurality of build machines employed in building a software system***. The single workstation embodiment disclosed in Boehm *et al.* is practiced on a standalone workstation. This system has only *one* build machine, not a plurality of build machines as recited in independent claim 1. For that reason alone, the single workstation embodiment of Boehm *et al.* does not anticipate claim 1. Furthermore, because there is only one build machine, the published files are not shared by a plurality of build machines.

In the Advisory Action, the Examiner contends that Boehm *et al.* teaches a build machine configuration that can be practiced on a standalone workstation. (See Advisory Action dated August 3, 2004 referencing the Final Office Action dated April 20, 2004 at pp. 15-16). The passage of Boehm *et al.* cited to by the Examiner states that:

[T]he present invention can be practiced with a single network cache that holds source and object files, or alternatively, one or more local caches, if the present invention is being practiced on a standalone workstation.

(See col. 9, lines 10-14). The Examiner contends that “the aspect of the invention disclosed by Boehm *et al.* consisting of a single workstation encompassing the components does not depart from the applicant’s claimed invention, as claim 2 of the applicant’s invention states ‘wherein the first component, the second component and the one or more build machines execute on a single computer.’ ” (Final Office Action at pp. 15-16). In making this argument, the Examiner ignores that claim 2 depends from claim 1 and as such contains all the limitations of claim 1.

As discussed above, claim 1 recites a first component for building a list of file names of published files to be ***shared by a plurality of build machines*** employed in building a software system. Thus, as illustrated by claim 2, there is a distinction between *build machines* and the *computer/workstation* upon which the one or more build machines are resident. Even though claim 2 requires that all the build machines execute on a single computer, because it depends from claim 1, claim 2 still recites the limitation that there be a plurality of build machines that share the published files. For example, in the system recited in claim 2, the plurality of build machines may be a plurality of virtual build machines executing on a single computer. (See *e.g.*, Application at p. 9, lines 6-8). In contrast, the single workstation embodiment of Boehm *et al.* has only a *single* build machine, *not a plurality* of build machines. Therefore the single workstation embodiment of Boehm *et al.* does not teach the ***sharing of published files by a plurality of build machines*** as in applicant’s claimed invention.

Anticipation requires that the ***identical*** invention be disclosed in a single reference. (See *e.g.*, *Richardson*, 868 F.2d at 1236, 9 U.S.P.Q.2d at 1920). Neither of the embodiments disclosed in Boehm *et al.* have all of the limitations of claim 1 together *in one system*. Nowhere in Boehm *et al.* is there a teaching of having *both* a first component for building a list of file names of published files to be ***shared by a plurality of build machines*** employed in building the software system *and* a second component for distributing to one or more build machines one or more published files, identified in the list of file names, that are to be ***stored persistently by the one or more build machines***. Therefore, it is readily apparent that the cited reference does not anticipate applicant’s invention as recited in independent claim 1 (and the claims that respectively depend there from).

Accordingly, the rejection of claims 1-4 and 7-13 should be withdrawn and allowance of the subject claims is respectfully requested

ii. **Claims 15-18**

Claims 15-18 stand rejected under 35 U.S.C. §102(e) as being anticipated by Boehm *et al.* Reversal of this rejection is respectfully requested for at least the following reasons. Boehm *et al.* does not disclose each and every limitation of these claims. In particular, the multi-workstation embodiment of Boehm *et al.* does not have a component for broadcasting to one or more of the build machines one or more published build files that are ***stored persistently by a set of the one or more build machines***. The single workstation embodiment of Boehm *et al.* does not disclose these features as recited in applicant's claimed invention.

Neither of the embodiments disclosed in Boehm *et al.* have all the limitations of claim 15 together *in one system*. Independent claim 15 recites *a system having a component for broadcasting to one or more of the build machines one or more published build files that are stored persistently by a set of the one or more build machines*. As explained in Section VII(A)(i) *supra*, the multi-workstation embodiment of Boehm *et al.* does not teach persistently storing build files *on* build machines. Rather, the files are stored in a network cache memory that is not on the build machines. For this reason alone, the single workstation embodiment of Boehm *et al.* does not anticipate claim 15.

Furthermore, because it has only one build machine, the single workstation embodiment of Boehm *et al.* does not broadcast to one or more of the build machines one or more published build files as in applicant's claimed invention. In accordance with the invention as recited in the subject claims, build files are stored on the build machines and are broadcast to the build machines by other build machines. (See *e.g.*, Application at p.8, lines 18-21 and 26-30; p. 9, lines 8-9; p. 10, lines 14-21; and p. 11, line 26 through p. 12, line 8; p. 14, lines 4-15 and 25-41). The single workstation embodiment taught by Boehm *et al.* consists of a software building program (*e.g.*, MAKE) running on a single workstation utilizing a local cache. (See *e.g.*, Boehm *et al.* at col. 9, lines 12-14). Boehm *et al.* does not expressly teach how files stored in the local cache are accessed. If files are stored in the local cache, presumably, there would be no need for a cache link structure

because all the files are stored locally. Regardless of how the files in the local cache are accessed, Boehm *et al.* does not disclose or suggest that published build files are broadcast to one or more build machines by other build machines.

In view of at least the foregoing, it is readily apparent that Boehm *et al.* does not anticipate applicant's invention as recited in independent claim 15 (and claims 16-18 which respectively depend there from). Accordingly, this rejection should be withdrawn.

iii. Claims 19 and 20

Claims 19 and 20 stand rejected under 35 U.S.C. §102(e) as being anticipated by Boehm *et al.* Reversal of this rejection is respectfully requested for at least the following reasons. Boehm *et al.* fails to disclose each and every limitation of the claims. In particular, the multi-workstation embodiment of Boehm *et al.* does not disclose methods or computer-executable instructions for sending one or more build files to a ***post build machine, and persistently storing the one or more build files on the one or more build machines*** as recited in the subject claims. Therefore, the invention as recited in the subject claims is not anticipated by Boehm *et al.*

More particularly, both claims 19 and 20 recite sending the one or more build files to the ***post-build machine***. Nowhere in Boehm *et al.* is a post-build machine disclosed either expressly or inherently. Additionally, the multi-workstation embodiment of Boehm *et al.* also does not disclose ***persistently storing build files on the build machines***.

In addition to lacking a post-build machine, the single workstation embodiment of Boehm *et al.* also does not distribute build files to the build machines. Both claims 19 and 20 recite the limitation ***distributing*** one or more build files to the one or more build machines, and persistently storing the one or more build files ***distributed*** to the one or more build machines. In the invention as recited in the claims, build files are stored on the build machines and are distributed to the build machines by other build machines or by the post-build machine. (See *e.g.*, Application at p.7, lines 18-23 and 27-30; p. 8, line 2 and lines 29-30; p. 9, lines 23-24 and lines 26-31; p. 10, lines 8-10; p. 11, lines 4-15 and 25-41; and p. 12, lines 1-7 and 25-26). As noted *supra*, the single workstation embodiment taught by Boehm *et al.* consists of a software building program (*e.g.*,

MAKE) running on a single workstation utilizing a local cache. (See e.g., Boehm *et al.* at col. 9, lines 12-14). Boehm *et al.* does not expressly teach how files stored in the local cache are accessed. If files are stored in the local cache, presumably, there would be no need for a cache link structure because all the files are stored locally. Regardless of how the files in the local cache are accessed, Boehm *et al.* does not teach that build files are distributed to one or more build machines by other build machines or by a post-build machine.

In view of at least the foregoing, it is submitted that Boehm *et al.* does not anticipate applicant's invention as recited in claims 19 and 20; and this rejection should be withdrawn.

iv. **Claims 21-23**

Claims 21-23 stand rejected under 35 U.S.C. §102(e) because the Examiner contends that these claims are unpatentable over Boehm *et al.* Reversal of this rejection is respectfully requested for at least the following reasons. Neither embodiment taught by Boehm *et al.* discloses each and every limitation of the claims. In particular, the multi-workstation embodiment of Boehm *et al.* does not disclose methods for **collecting** one or more build files **from one or more build machines**. Likewise, the single workstation embodiment of Boehm *et al.* does not disclose methods for collecting one or more build files from one or more build machines *and* distributing the one or more build files to the one or more build machines.

Claim 21 recites the limitations **collecting** one or more build files **from one or more build machines** and **distributing** the one or more build files **to the one or more build machines**. Neither of the embodiments disclosed in Boehm *et al.* have all of the limitations of claim 21 *together* in one system. In the multi-workstation embodiment of Boehm *et al.*, the files are not collected from one or more build machines. As explained in Section VII(A)(i) *supra*, the multi-workstation embodiment taught by Boehm *et al.* stores files in a network cache memory that is not on a build machine. Because the network cache memory is not on a build machine, a system that collects a file from a network cache memory is not collecting a file from a build machine. Therefore, the

multi-workstation embodiment of Boehm *et al.* does not teach collecting one or more build files from one or more build machines as in applicant's claimed invention.

In view of the above, it is clear that Boehm *et al.* does not anticipate applicant's invention as recited in claim 21 (and claims 22 and 23 which depend there from). Accordingly, this rejection should be withdrawn and allowance of the subject claims is respectfully requested.

v. **Claim 24**

Regarding claim 24, the multi-workstation embodiment of Boehm *et al.* does not disclose computer-executable instructions for *collecting* one or more build files *from one or more build machines, and for persistently storing* transmitted files as recited in this claim. As noted *supra*, neither of the embodiments taught by Boehm *et al.* *both collect* one or more build files *from one or more build machines, and distribute* the one or more build files to the one or more build machines. For that reason alone, the teachings of Boehm *et al.* do not anticipate claim 24. Additionally, claim 24 also recites the limitation that transmitted files are *persistently stored*. In the invention as recited in the claims, the transmitted files are persistently stored *on the build machines*. As discussed above, the multi-workstation embodiment of Boehm *et al.* does not store the build files on the build machines, but rather, stores the build files in a network cache memory, which is not on the build machine. Accordingly, Boehm *et al.* does not anticipate applicant's invention as recited in the subject claim, and this rejection should be withdrawn

vi. **Claim 25**

Regarding claim 25, Boehm *et al.* does not disclose a *post-build machine* as recited in this claim. In addition, the multi-workstation embodiment of Boehm *et al.* does not have means for *persistently storing* one or more build files *distributed to the one or more build machines*, and the single workstation embodiment of Boehm *et al.* does not have means for *distributing* one or more build files to the one or more build machines as in applicant's claimed invention.

Claim 25 recites the limitation means for sending the one or more build files to a *post-build machine*. Nowhere in Boehm *et al.* is a post-build machine disclosed and,

accordingly, nowhere in Boehm *et al.* is a means for sending one or more build files to a post-build machine disclosed. Thus, applicant's invention as recited in claim 25 is not disclosed by Boehm *et al.* Accordingly, this rejection should be withdrawn.

B. Rejection of Claim 14 Under 35 U.S.C. §103(a)

Claim 14 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Boehm *et al.* in view of Lubkin *et al.* Reversal of this rejection is respectfully requested for at least the following reasons. There is no teaching, suggestion, or motivation to combine Boehm *et al.* with Lubkin *et al.* and the Examiner has not provided any *evidence in the prior art or other reference of record* to show otherwise. It appears the Examiner has impermissibly employed teachings of applicant's specification as a 20/20 hindsight-based roadmap to achieve the purported combination.

The Examiner incorrectly contends that:

[I]t would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the hashing data structure as disclosed by Lubkin *et al.* with the system for building software of Boehm *et al.*, as this would allow the system to quickly search the data structure containing file names by resolving hash values in the system disclosed by Boehm *et al.*

(Final Office Action at p. 15). However, the Examiner does not point to any passages in either Boehm *et al.* or Lubkin *et al.* that disclose this purported motivation to combine the references. Instead, the Examiner is basing the rejection on the assertion that it would have been obvious to do something not suggested in the art but, rather, suggested by the advantages disclosed in the applicant's specification. This sort of approach has been repeatedly condemned by the Federal Circuit as entering the "tempting but forbidden zone of hindsight." (*In re Dembiczak*, 175 F.3d at 998, 50 U.S.P.Q. at 1616 (citing to *Loctite Corp. v. Ultraseal Ltd.*, 781 F.2d 861, 873, 228 U.S.P.Q. 90, 98 (Fed. Cir. 1985), overruled on other grounds by *Nobelpharma AB v. Implant Innovations, Inc.*, 141 F.3d 1059, 46 U.S.P.Q.2d 1097 (Fed. Cir. 1998))). By imbuing "one of ordinary skill in the

art with knowledge of the invention in suit when *no prior art reference or references of record* convey or suggest knowledge,” the Examiner has fallen “victim to the insidious effect of a hindsight syndrome wherein that which only the inventor taught is used against its teacher.” (*In re Fine*, 837 F.2d 1071, 1075, 5 U.S.P.Q.2d 1596, 1600 (Fed. Cir. 1988) (citing to *W.L. Gore & Assoc., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1553, 220 U.S.P.Q. 303, 312-13 (Fed. Cir. 1983)) (emphasis added); *see also In re Dembiczak*, 175 F.3d at 998-99, 50 U.S.P.Q. at 1616-17). As the Federal Circuit has stated:

Measuring a claimed invention against the standard established by section 103 requires the oft-difficult but critical step of casting the mind back to the time of invention, to consider the thinking of one of ordinary skill in the art, guided only by the prior art references and the then-accepted wisdom in the field. . . . Our case law makes clear that the best defense against the subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references. Combining prior art references *without evidence* of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability--the essence of hindsight. . . . [T]he showing must be *clear and particular*. ***Broad conclusory statements regarding the teaching of multiple references, standing alone, are not "evidence."***

(*In re Dembiczak*, 175 F.3d at 999, 50 U.S.P.Q. at 1617) (internal citations omitted) (emphasis added). Indeed, all the Examiner has done is provide the “[b]road conclusory statements” that the Federal Circuit has held are not *evidence* that can support an obviousness rejection. (*Id.*). Accordingly, absent impermissible 20/20 hindsight based reliance on applicant’s specification, the Examiner has failed to provide any motivation to effect the purported combination of references. In view of at least the foregoing, it is respectfully submitted that the Examiner has failed to make an adequate showing of a teaching, suggestion, or motivation to combine the references, and thus, has failed to establish a *prima facie* case of obviousness.

Moreover, for at least the reasons discussed in Section VII(A)(i) *supra*, Boehm *et al.* does not teach each and every limitation of independent claim 1. Claim 14 depends

from this claim, and accordingly, claim 14 is allowable for at least the same reasons noted above regarding claim 1 since Lubkin *et al.* fails to cure the aforementioned deficiencies of Boehm *et al.* with respect this independent claim. Hence, this rejection should be withdrawn and allowance of claim 14 is respectfully requested.

C. Conclusion

For at least the above reasons, the claims currently under consideration are believed to be patentable over the cited references. Accordingly, it is respectfully requested that the rejections of claims 1-4 and 7-25 be reversed.

If any additional fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Respectfully submitted,
AMIN & TUROCY, LLP



Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24th Floor, National City Center
1900 East 9th Street
Telephone: (216) 696-8730
Facsimile: (216) 696-8731

VIII. Claims Appendix (37 C.F.R. §41.37(c)(1)(viii))

1. A system for building a software system, the building system comprising:
a first component for building a list of file names of published files to be shared by a plurality of build machines employed in building the software system; and
a second component for distributing to one or more of the build machines one or more published files, identified in the list of file names, that are to be stored persistently by the one or more build machines.
2. The system of claim 1, wherein the first component, the second component and the one or more build machines execute on a single computer.
3. The system of claim 1, wherein the first component, the second component and the one or more build machines execute on a plurality of computers.
4. The system of claim 1, wherein each of the first component, the second component and the one or more build machines execute on separate computers.
- 5-6. Withdrawn
7. The system of claim 1, stored on a computer readable medium.
8. The system of claim 1, wherein the list of published file names of files is a list of pathnames.
9. The system of claim 1, wherein the second component initiates file transfers within the system for building a software system.
10. The system of claim 1, wherein one or more of the build machines initiates file transfers within the system for building a software system.

11. The system of claim 1, wherein a process other than the first component, the second component or the one or more build machines initiates file transfers within the system for building a software system.
12. The system of claim 1, wherein the first component builds a list of updates to files to be shared by the plurality of build machines and the second component distributes to one or more of the build machines updates to one or more files that are stored persistently by the one or more build machines.
13. The system of claim 1, further comprising a first data structure for storing the list of published file names.
14. The system of claim 13, wherein the data structure is a hash.
15. A system for building a software system employing a plurality of build machines, the building system comprising:
 - a component for broadcasting to one or more of the build machines one or more published build files that are stored persistently by a set of the one or more build machines; and
 - a component for selectively receiving and storing persistently one or more of the broadcast published build files.
16. The system of claim 15, wherein the component for broadcasting the one or more published build files and the one or more build machines execute on one computer.
17. The system of claim 15, wherein the component for broadcasting the one or more published build files and the one or more build machines execute on a plurality of computers.

18. The system of claim 15, wherein each of the components for broadcasting the one or more published build files and the one or more build machines execute on separate computers.

19. A method for building a software system comprising:
collecting from one or more build machines, one or more file names
corresponding to the one or more build files;
determining which of the one or more build files the one or more build machines
are to send to a post build machine;
persistently storing the one or more build files on the one or more build machines;
sending to the one or more build machines a list of file names of build files to be
sent to the post build machine;
sending the one or more build files to the post build machine;
determining which of the one or more build files the one or more build machines
are to receive from the post build machine;
sending to the one or more build machines a list of file names of build files to be
received from the post build machine;
distributing one or more build files to the one or more build machines; and
persistently storing the one or more build files distributed to the one or more build
machines.

20. A computer-readable medium having stored thereon computer-executable instructions comprising:
- collecting from one or more build machines, one or more file names corresponding to the one or more build files;
 - determining which of the one or more build files the one or more build machines are to send to a post build machine;
 - persistently storing the one or more build files on the one or more build machines;
 - sending to the one or more build machines a list of file names of build files to be sent to the post build machine;
 - sending the one or more build files to the post build machine;
 - determining which of the one or more build files the one or more build machines are to receive from the post build machine;
 - sending to the one or more build machines a list of file names of build files to be received from the post build machine;
 - distributing one or more build files to the one or more build machines; and
 - persistently storing the one or more build files distributed to the one or more build machines.
21. A method for building a software system comprising:
- collecting one or more build files from one or more build machines; and
 - distributing the one or more build files to the one or more build machines.
22. The method of claim 21, wherein distributing the one or more build files further comprises:
- broadcasting the one or more build files to the one or more build machines.
23. The method of claim 22, wherein collecting the one or more build files from the one or more build machines further comprises:
- determining which of the broadcast build files to store persistently; and
 - persistently storing one or more of the broadcast files.

24. A computer-readable medium having stored thereon computer-executable instructions comprising:

- collecting one or more build files from one or more build machines;
- distributing the one or more build files to the one or more build machines;
- determining which of the transmitted build files to store persistently; and
- persistently storing one or more of the transmitted files.

25. A system for building a software system comprising:

- means for collecting from one or more build machines, one or more file names corresponding to the one or more build files;
- means for determining which of the one or more build files the one or more build machines are to send to a post build machine;
- means for sending the one or more build files to the post build machine;
- means for determining which of the one or more build files the one or more build machines are to receive from the post build machine;
- means for sending to the one or more build machines a list of file names of build files to be sent to the one or more build machines and the build files to be received from the one or more build machines;
- means for distributing one or more build files to the one or more build machines;
- and
- means for persistently storing one or more build files distributed to the one or more build machines.

26-35. Withdrawn

IX. Evidence Appendix (37 C.F.R. §41.37(c)(1)(ix))

None.

X. Related Proceedings Appendix (37 C.F.R. §41.37(c)(1)(x))

None.